

Appendix 2 – Bayesian Parameterization of models

A2.1 Temperature Model

Below is the temperature model, constructed for a WinBUGS analysis. Code annotations appear after hashmarks(#). It is critical to note that Bayesian programs such as WinBUGS and JAGS do not process items in a linear order. The model and all components are run simultaneously. For this reason, an item defined at the top of the program is not estimated “first.”

The temperature model includes for a correlation between the slope (Control) and intercept random effects. Random effects are by site and are estimated as bivariate normal ($\mu = 0$).

Anything in the model that is followed by a [~] represents a parameter distribution for WinBUGS or JAGS to estimate (stochastic node). Anything followed by [<–] is a logical expression (logical node), likely containing manipulations of estimated parameters.

```
model {  
  
## Setting Priors  
for (i in 1:ngrups)    #33 groups (sites)  
{  
    alpha[i] <- B[i,1]  #Site intercept  
    betal[i] <- B[i,2]  #Site slope, by control reach temp  
    B[i,1:2] ~ dmnorm(B.hat[i,], Tau.B[,])  
        #multivariate (bivariate) normal distribution priors with per-site  
        #values of i, precision Tau  
    B.hat[i,1] <- mu.int      #fixed effect: intercept  
    B.hat[i,2] <- mu.betal    #fixed effect: for Control slope  
}  
  
mu.int ~ dnorm(0, 0.001)      # Hyperpriors for random intercepts  
mu.betal ~ dnorm(0, 0.001)    # Hyperpriors for random slopes (Control)  
  
Tau.B[1:2,1:2] <- inverse(Sigma.B[,])      #constructing the random effects  
Sigma.B[1,1] <- pow(sigma.int,2)            #variance-covariance matrix  
sigma.int ~ dunif(0, 100)                    # SD of random intercepts, random uniform  
Sigma.B[2,2] <- pow(sigma.betal,2)  
sigma.betal ~ dunif(0, 100)                  # SD of random slopes, random uniform  
Sigma.B[1,2] <- rho*sigma.int*sigma.betal  
Sigma.B[2,1] <- Sigma.B[1,2]  
rho ~ dunif(-1,1)                          #residual “random” effects  
covariance <- Sigma.B[1,2]  
  
beta2 ~ dnorm(0, 0.001)      #Fixed effect: TRLength  
beta3 ~ dnorm(0, 0.001)      #Fixed effect: Shade  
beta4 ~ dnorm(0, 0.001)      #Fixed effect: GradlQ  
  
tau <- 1 / ( sigma * sigma)      # Residual  
sigma ~ dunif(0, 100)          # Residual standard deviation  
  
# Likelihood  
# n= number of temperature measurements (119)
```

```

for (i in 1:n) {
  response[i] ~ dnorm(mu[i], tau) # The 'residual' random variable, with
  #a mean of mu (below) and precision tau (above)
  mu[i] <- alpha[site[i]] + betal[site[i]]* c_ControlTemp[i] +
  beta2*c_TReachLength[i] + beta3*c_Shade[i] + beta4*c_Grad1Q[i]
  #mu = equation 1. It is the expectation or predicted result given
  #parameter estimates and data values
}
}

# Initial values for model to operate with
inits <- function(){ list(mu.int = rnorm(1, 0, 1), sigma.int = rlnorm(1),
mu.betal = rnorm(1, 0, 1), sigma.betal = rlnorm(1), rho = runif(1, -1, 1),
sigma = rlnorm(1),
beta2=rnorm(1), beta3=rnorm(1), beta4=rnorm(1)) }

# Parameters to provide estimates of
parameters <- c("alpha", "betal", "beta2", "beta3", "beta4", "mu.int",
"sigma.int", "mu.betal",
"sigma.betal", "rho", "covariance", "sigma")

# MCMC settings
ni <- 4000 #number of iterations
nb <- 2000 #number for burn-in (toss out first 2000 iterations)
nt <- 2      #Thinning: don't use every other value
nc <- 6      #Number of chains

# Start Gibbs sampler
out <- bugs(win.data, inits, parameters, "lme.GradShade.txt", n.thin=nt,
n.chains=nc, n.burnin=nb, n.iter=ni, debug = TRUE)

```

A2.2 Shade Model

The Bayesian formulation of the shade model is provided below. The model was run in JAGS. Code annotations appear after hashmarks(#).

The shade model is a weighted regression. Note that the values for Logit_shade_var provided for fixed sampling variance, used in constructing the weighted regression. The variables
PctDiffBAlt100 = Percent Difference in Basal Area, less than 100' from stream
Pre100PctHWD = Percent hardwood within 100' of stream, pre-harvest
TreeHt100m = Tree height pre-harvest, within 100' of stream

Anything in the model that is followed by a [~] represents a parameter distribution for WinBUGS or JAGS to estimate (stochastic node). Anything followed by [<-] is a logical expression (logical node), likely containing manipulations of estimated parameters.

Within JAGS, we specified 20,000 iterations with 6 chains. By default, JAGS will discard the first 50% of iteration estimates and use enough to obtain 1000 estimates (1 in 10 thinning in this case).

```

#####SHADE#####
##priors for shade (suffix=S)
alphaS ~ dnorm (0, 0.001)
beta1S ~ dnorm(0, 0.001)
beta2S ~ dnorm(0, 0.001)
beta3S ~ dnorm(0, 0.001)
idelS ~ dunif(0, 100) # Residual
for (i in 1:n.sites)
{
  for (j in 1:2) #j = pre or post-harvest
    {tauSh[i,j]<-1/Logit_shade_var[i,j] #fixed variance (for weights)
    }
}
#### Likelihood
for (i in 1:n.sites)
{
  muS[i,1]<-LogitShade[i,1] ##This is pre-harvest shade, using raw values

  Sh_post[i,2] ~ dnorm(LogitShade[i,2],idelS) # The 'residual' random
  ##variable, process variance (idelSh = "i-delta-shade")
  LogitShade[i,2] ~ dnorm(muS[i,2], tauSh[i,2]) #setting up a fixed
  ##(weighted) sampling variance
  muS[i,2]<-alphaS + beta1S*PctDiffBAlt100[i] + beta2S*Pre100PctHWD[i] +
  beta3S*TreeHt100m[i] ## muS = predicted shade value given model

}

##parameters for estimation
jags.params <- c("alphaS", "beta1S", "beta2S", "beta3S", "idelS")

##initial parameter values (randomized)
jags.inits<- function(){ list(alphaS = rnorm(1), beta1S = rnorm(1),
beta2S=rnorm(1), beta3S=rnorm(1), idelS = rlnorm(1))}
```

A2.3 Combined Model

The combined model consists of the temperature model from A2.1 and shade model from A2.2. However, there is a (logistic) transformation between the logit of shade values and the values for shade entered into the temperature sub-model here. Shade parameters have their own postscript, S (e.g., alphaS = intercept parameter for shade). Commentary on many parameters is not included below; see A2.1 and A2.2.

For predicting stream temperature changes, the (derived) variables `delta_sim`, `delta_muT_sim`, `estTemp[i]`, and `estTempNE[i]` are key. `Delta_sim` is the across-site mean of all predicted temperature change. `Delta_muT_sim` is a per-site estimate, calculated by subtracting the two harvest scenarios `estTemp` (harvest) and `estTempNE` (NE = no effect, or no harvest). The simulated value for Percent Difference in Basal Area is `PctDiffBAsim`.

Following priors, the shade model is followed by prediction steps (relying on temperature parameter distribution estimates) and finally the temperature model. Running this model using

JAGS with 600,000 iterations appeared sufficient to reach a stable answer (300,000 iterations discarded as burn-in, 1 in 300 retained for estimates).

```

##### Priors #####
## Priors for temperature
for (i in 1:n.sites){
  alpha[i] <- B[i,1]                      #Site intercept
  beta1[i] <- B[i,2]  #Site slope, by control reach temperature
  B[i,1:2] ~ dmnorm(B.hat[i,], Tau.B[,])
  B.hat[i,1] <- mu.int
  B.hat[i,2] <- mu.betal
}

mu.int ~ dnorm(0, 0.001)      # Hyperpriors for random intercepts
mu.betal ~ dnorm(0, 0.001)    # Hyperpriors for random slopes (control
                             # reach temp)

Tau.B[1:2,1:2] <- inverse(Sigma.B[,])
Sigma.B[1,1] <- pow(sigma.int,2)
sigma.int ~ dunif(0, 100)      # SD of intercepts
Sigma.B[2,2] <- pow(sigma.betal,2)
sigma.betal ~ dunif(0, 100)    # SD of slopes
Sigma.B[1,2] <- rho*sigma.int*sigma.betal
Sigma.B[2,1] <- Sigma.B[1,2]
rho ~ dunif(-1,1)
covariance <- Sigma.B[1,2]

beta2 ~ dnorm(0, 0.001)
beta3 ~ dnorm(0, 0.001)
beta4 ~ dnorm(0, 0.001)
tau <- 1 / ( sigma * sigma)    # Residual
sigma ~ dunif(0, 100)          # Residual standard deviation

##priors for shade (suffix=S)
alphas ~ dnorm (0, 0.001)      #shade intercept
betals ~ dnorm(0, 0.001)
beta2S ~ dnorm(0, 0.001)
beta3S ~ dnorm(0, 0.001)
idelS ~ dunif(0, 100)          # Residual random variable

for (i in 1:n.sites)
{
  for (k in 1:2)
    {taush[i,k]<-1/Logit_shade_var[i,k]  #Fixed shade variance assignment by
     # pre/post harvest
    }
}
delta_sim<-mean(delta_muT_sim)### Mean of the simulated temperature increase
                         ### across sites

##### Likelihood #####
for (i in 1:n.sites)

```

```

{
  muS[i,1]<-LogitShade[i,1] #Pre-harvest value for shade uses measured
  #values
  InvLogitMuS[i,1]<-(1/(exp(-(muS[i,1]+2.144418834))+1)) - 0.882449112
## Inverse of the logit. Both the logit of shade values and shade values
## are centered. The 2.144 value un-centers the logit of shade values.
## The 0.8824 value centers the resulting shade (between 0 and 1) value.

# Post-harvest shade estimation #
Sh_post[i,2] ~ dnorm(LogitShade[i,2],idelS) #idelS= The 'residual' random
# variable, process variance (idelSh = i-delta-shade)
LogitShade[i,2] ~ dnorm(muS[i,2], tauSh[i,2] ) #Logit of shade values (real)
# modeled as having a mean of MuS and fixed sampling variance tauSh (real,
# see above)

muS[i,2]<-alphaS + betalS*PctDiffBAlt100[i] + beta2S*Pre100PctHWD[i] +
beta3S*TreeHt100m[i] #MuS = equation describing mean response

InvLogitMuS[i,2]<-(1/(exp(-(muS[i,2]+1.774318915))+1)) - 0.833159815
# needed to alter the estimated shade value, this time for post-
# harvest. Numeric values differ because centering values differed
# between pre-harvest and post-harvest

#change in temperature between the estimated temp and the estimated temp with
no harvest effect (NE)

delta_muT_sim[i]<- estTemp[i]- estTempNE[i] #site-level predicted increase in
#temperature. See below.

#estimated temperatures from simulated harvest

muS_calc[i]<-alphaS + betalS*PctDiffBAsim[i] + beta2S*Pre100PctHWD[i] +
beta3S*TreeHt100m[i] #estimated shade value given PctDiffBAsim and other
# variables

InvLogitMuS_calc[i]<-(1/(exp(-(muS_calc[i]+1.774318915))+1)) - 0.833159815

#now sticking the estimated shade into the temperature model
estTemp[i]<-alpha[site[i]] + betal[site[i]]*C_temp[i,1,2] +
beta2*TReachLength[i] +
beta3*(InvLogitMuS_calc[i]) + beta4*Grad_1stQuart[i]

#estimated temperatures from no harvest

#estTempNE is for the estimated temparture, no harvest effect. The harvest
# effect is replaced by zero minus the centering amount for PctDiffBAlt100,
# 0.193039105

#Estimated NE temp shade transformation
muS_calcNE[i]<-alphaS + betalS*(-0.193039105) + beta2S*Pre100PctHWD[i] +
beta3S*TreeHt100m[i]

```

```

InvLogitMuS_calcNE[i]<- (1/(exp(-(muS_calcNE[i]+1.774318915))+1)) -
0.833159815

#now sticking the estimated no-effects shade into the temperature model
estTempNE[i]<-alpha[site[i]] + betal[site[i]]*C_temp[i,1,2] +
beta2*TReachLength[i] + beta3*(InvLogitMuS_calcNE[i]) +
beta4*Grad_1stQuart[i]

#### Estimating temperature parameters ####
for(k in 1:2)
{
  for(j in 1:n.instances[i,k])
  {

T_temp[i,j,k] ~ dnorm(mu[i,j,k], tau)      # The 'residual' random variable

mu[i,j,k] <- alpha[site[i]] + betal[site[i]]*C_temp[i,j,k] +
beta2*TReachLength[i] + beta3*InvLogitMuS[i,k] + beta4*Grad_1stQuart[i]

}
}
}

## parameter values for reporting
jags.params <- c("beta2", "beta3", "beta4", "mu.int", "sigma.int",
"mu.betal", "sigma.betal", "rho", "covariance", "sigma", "alphas", "betals",
"beta2S", "beta3S", "idelS", "delta_muT_sim", "delta_sim")

## initial values
jags.inits<- function(){ list(mu.int = rnorm(1, 0, 1), sigma.int = rlnorm(1),
mu.betal = rnorm(1, 0, 1), sigma.betal = rlnorm(1), rho = runif(1, -1, 1),
sigma = rlnorm(1), beta2=rnorm(1), beta3=rnorm(1), beta4=rnorm(1), alphas =
rnorm(1), betals = rnorm(1), beta2S=rnorm(1), beta3S=rnorm(1), idelS =
rlnorm(1)) }

```

A2.4 As-harvested Parameter Estimates

Below are listed the parameter estimates from the As-harvested model presented in A2.3. Individual random effects estimates from the temperature sub-model are listed as alpha[] and beta[]. All parameters are alphabetically listed.

	mean	sd	2.50%	25%	50%	75%	97.50%
alpha[1]	0.549	0.374	-0.197	0.291	0.562	0.779	1.326
alpha[2]	0.693	0.199	0.303	0.556	0.695	0.825	1.088
alpha[3]	1.234	0.278	0.696	1.048	1.224	1.441	1.772
alpha[4]	1.427	0.278	0.867	1.241	1.427	1.618	1.970
alpha[5]	0.419	0.314	-0.208	0.209	0.402	0.627	1.032
alpha[6]	0.101	0.256	-0.429	-0.056	0.102	0.272	0.579
alpha[7]	0.664	0.235	0.194	0.503	0.668	0.816	1.110
alpha[8]	1.084	0.214	0.675	0.939	1.085	1.235	1.496
alpha[9]	-0.341	0.377	-1.052	-0.613	-0.345	-0.069	0.342
alpha[10]	0.104	0.216	-0.308	-0.053	0.097	0.255	0.534

alpha[11]	0.825	0.257	0.342	0.649	0.817	0.994	1.314
alpha[12]	-1.880	0.199	-2.251	-2.018	-1.885	-1.743	-1.471
alpha[13]	0.267	0.197	-0.107	0.136	0.264	0.403	0.663
alpha[14]	0.460	0.312	-0.191	0.261	0.460	0.663	1.060
alpha[15]	0.816	0.199	0.448	0.674	0.810	0.949	1.200
alpha[16]	-0.141	0.346	-0.841	-0.365	-0.123	0.088	0.485
alpha[17]	0.529	0.238	0.080	0.374	0.521	0.674	1.015
alpha[18]	0.665	0.388	-0.089	0.414	0.660	0.904	1.417
alpha[19]	-0.204	0.542	-1.291	-0.561	-0.210	0.138	0.916
alpha[20]	0.016	0.207	-0.417	-0.119	0.024	0.152	0.402
alpha[21]	0.667	0.214	0.244	0.524	0.673	0.820	1.085
alpha[22]	-0.216	0.487	-1.173	-0.535	-0.202	0.103	0.747
alpha[23]	0.354	0.323	-0.264	0.138	0.353	0.567	1.013
alpha[24]	-0.002	0.328	-0.637	-0.231	0.004	0.217	0.642
alpha[25]	0.233	0.413	-0.559	-0.022	0.223	0.482	1.062
alpha[26]	1.162	0.262	0.658	0.986	1.155	1.335	1.679
alpha[27]	0.871	0.353	0.172	0.634	0.860	1.104	1.559
alpha[28]	1.125	0.222	0.697	0.978	1.125	1.271	1.568
alpha[29]	-0.304	0.383	-1.091	-0.565	-0.289	-0.037	0.425
alpha[30]	-0.016	0.279	-0.567	-0.196	-0.014	0.167	0.544
alpha[31]	0.861	0.264	0.340	0.683	0.855	1.049	1.377
alpha[32]	0.677	0.264	0.130	0.506	0.682	0.843	1.178
alpha[33]	0.385	0.267	-0.192	0.216	0.395	0.566	0.894
alphaS	-0.279	0.066	-0.407	-0.321	-0.279	-0.237	-0.148
beta1[1]	-0.380	1.839	-3.681	-1.605	-0.486	0.769	3.655
beta1[2]	-0.750	1.742	-4.098	-1.928	-0.803	0.403	2.756
beta1[3]	-2.908	0.579	-4.039	-3.300	-2.912	-2.528	-1.805
beta1[4]	0.044	0.719	-1.428	-0.426	0.065	0.540	1.342
beta1[5]	-0.015	1.403	-2.595	-0.974	-0.100	0.906	3.010
beta1[6]	-0.651	1.726	-3.916	-1.852	-0.677	0.468	2.779
beta1[7]	-0.499	0.439	-1.327	-0.794	-0.512	-0.190	0.389
beta1[8]	-1.268	1.799	-4.727	-2.428	-1.217	-0.089	2.430
beta1[9]	-2.129	0.599	-3.310	-2.532	-2.116	-1.716	-0.985
beta1[10]	-0.656	1.933	-4.170	-1.962	-0.779	0.553	3.490
beta1[11]	0.714	1.840	-2.583	-0.508	0.611	1.899	4.707
beta1[12]	-2.806	1.026	-4.938	-3.451	-2.828	-2.156	-0.688
beta1[13]	-3.116	0.475	-4.063	-3.435	-3.134	-2.799	-2.141
beta1[14]	-3.602	0.907	-5.355	-4.185	-3.580	-2.961	-1.907
beta1[15]	-0.933	0.905	-2.617	-1.573	-0.938	-0.315	0.902
beta1[16]	-0.993	1.722	-4.316	-2.104	-1.075	0.094	2.378
beta1[17]	-1.088	1.660	-4.385	-2.166	-1.088	0.109	2.090
beta1[18]	-1.532	0.714	-3.041	-1.985	-1.509	-1.094	-0.191
beta1[19]	-0.206	1.219	-2.724	-0.985	-0.227	0.535	2.282
beta1[20]	-0.110	1.703	-3.161	-1.233	-0.249	0.908	3.578
beta1[21]	-0.029	0.935	-1.912	-0.649	-0.035	0.610	1.786

beta1[22]	-2.450	0.845	-4.140	-3.010	-2.412	-1.883	-0.826
beta1[23]	-2.396	1.366	-5.153	-3.249	-2.330	-1.521	0.335
beta1[24]	-1.230	0.843	-2.818	-1.813	-1.243	-0.664	0.467
beta1[25]	-1.332	1.336	-4.008	-2.161	-1.365	-0.479	1.374
beta1[26]	0.453	1.135	-1.876	-0.309	0.502	1.231	2.612
beta1[27]	-1.030	0.665	-2.344	-1.462	-1.012	-0.589	0.234
beta1[28]	-3.332	0.609	-4.525	-3.717	-3.355	-2.937	-2.132
beta1[29]	-1.962	1.850	-5.665	-3.159	-1.949	-0.794	1.630
beta1[30]	-0.462	1.396	-3.087	-1.409	-0.501	0.447	2.317
beta1[31]	0.109	1.778	-3.150	-1.153	0.042	1.203	3.867
beta1[32]	-1.043	1.737	-4.561	-2.194	-1.022	0.025	2.431
beta1[33]	2.312	1.250	-0.064	1.493	2.286	3.162	4.711
beta1S	-2.776	0.305	-3.428	-2.973	-2.770	-2.558	-2.223
beta2	0.871	0.336	0.206	0.663	0.878	1.077	1.511
beta2S	-0.585	0.249	-1.092	-0.754	-0.583	-0.414	-0.100
beta3	-5.606	0.844	-7.341	-6.153	-5.590	-5.030	-4.046
beta3S	-0.065	0.017	-0.100	-0.076	-0.066	-0.054	-0.031
beta4	-0.077	0.049	-0.179	-0.109	-0.076	-0.044	0.014
covariance	0.218	0.351	-0.435	-0.011	0.211	0.429	0.954
delta_muT_sim[5]	0.300	0.045	0.219	0.267	0.300	0.330	0.392
delta_muT_sim[8]	1.069	0.116	0.855	0.984	1.068	1.148	1.291
delta_muT_sim[9]	1.309	0.148	1.033	1.200	1.308	1.414	1.594
delta_muT_sim[10]	2.109	0.209	1.715	1.967	2.108	2.255	2.499
delta_muT_sim[11]	1.152	0.113	0.923	1.074	1.153	1.227	1.370
delta_muT_sim[12]	0.448	0.051	0.354	0.411	0.448	0.484	0.544
delta_muT_sim[15]	-0.066	0.007	-0.081	-0.071	-0.066	-0.061	-0.052
delta_muT_sim[18]	0.303	0.055	0.207	0.264	0.300	0.339	0.421
delta_muT_sim[19]	0.306	0.044	0.222	0.276	0.306	0.335	0.394
delta_muT_sim[20]	1.134	0.136	0.856	1.044	1.133	1.227	1.394
delta_muT_sim[21]	0.974	0.095	0.785	0.909	0.975	1.037	1.151
delta_muT_sim[22]	1.331	0.163	1.006	1.218	1.332	1.438	1.652
delta_muT_sim[23]	0.358	0.041	0.278	0.330	0.358	0.385	0.439
delta_muT_sim[24]	2.196	0.219	1.769	2.048	2.195	2.346	2.606
delta_muT_sim[25]	2.173	0.213	1.763	2.026	2.172	2.312	2.569
delta_muT_sim[27]	0.212	0.023	0.169	0.196	0.211	0.227	0.257
delta_muT_sim[30]	0.591	0.072	0.457	0.539	0.590	0.639	0.731
delta_muT_sim[33]	0.913	0.102	0.717	0.843	0.910	0.981	1.114
delta_muT_sim[1]	-0.050	0.007	-0.065	-0.055	-0.050	-0.045	-0.036
delta_muT_sim[2]	-0.169	0.021	-0.209	-0.183	-0.168	-0.155	-0.130
delta_muT_sim[3]	-0.020	0.003	-0.026	-0.022	-0.020	-0.018	-0.015
delta_muT_sim[4]	0.036	0.004	0.028	0.033	0.035	0.038	0.044
delta_muT_sim[6]	-0.027	0.003	-0.034	-0.030	-0.027	-0.025	-0.021
delta_muT_sim[7]	-0.003	0.000	-0.004	-0.003	-0.003	-0.003	-0.002
delta_muT_sim[13]	-0.006	0.001	-0.008	-0.007	-0.006	-0.006	-0.005
delta_muT_sim[14]	0.116	0.012	0.093	0.107	0.115	0.125	0.139

delta_muT_sim[16]	0.089	0.011	0.067	0.081	0.090	0.097	0.112
delta_muT_sim[17]	-0.098	0.014	-0.124	-0.107	-0.097	-0.089	-0.072
delta_muT_sim[26]	0.054	0.006	0.043	0.050	0.054	0.058	0.065
delta_muT_sim[28]	-0.154	0.034	-0.226	-0.175	-0.151	-0.128	-0.097
delta_muT_sim[29]	-0.072	0.011	-0.094	-0.080	-0.072	-0.065	-0.053
delta_muT_sim[31]	-0.465	0.069	-0.602	-0.513	-0.464	-0.419	-0.327
delta_muT_sim[32]	-0.017	0.002	-0.021	-0.019	-0.017	-0.016	-0.013
delta_sim	0.486	0.049	0.395	0.451	0.485	0.519	0.576
deviance	85.727	17.489	54.325	73.848	84.686	96.386	122.888
idelS	49.967	28.579	2.444	25.738	48.646	75.080	96.816
mu.beta1	-1.092	0.460	-1.951	-1.399	-1.105	-0.807	-0.152
mu.int	0.396	0.133	0.104	0.310	0.400	0.484	0.645
rho	0.153	0.228	-0.297	-0.007	0.165	0.319	0.555
sigma	0.280	0.029	0.233	0.260	0.277	0.299	0.345
sigma.beta1	1.921	0.440	1.186	1.599	1.887	2.199	2.861
sigma.int	0.730	0.111	0.538	0.654	0.720	0.794	0.981